

EE 428: Computer Vision

Homework 0: Background Subtraction

Instructor: Jonathan Ventura

For this warm-up homework, you need to set up your Python environment. I recommended using [Visual Studio Code](#). You will also need to install NumPy, scikit-image and opencv-python. This is most easily done by creating a virtual environment and installing them with pip.

The goal of this assignment is to give you a chance to get your environment set up and practice the basics of working with images and video in Python with NumPy, scikit-image, and OpenCV. As an initial exercise, you will implement a simple "background subtraction" technique to separate foreground and background in a video.

You will submit your Python notebook and a short report (Word Doc or PDF) explaining your solution and answering the questions given below.

For various steps, such as converting from color to grayscale, applying Otsu's method, and putting boxes around blobs you will need to find the appropriate functions in scikit-image or OpenCV.

Code requirements

1. Load the frames, convert each one to grayscale, and store them in a NumPy array of shape $(N, \text{height}, \text{width})$. Show the frames as a video (use the `show_video` function from Lab 1.1).
2. Compute the per-pixel average of the video. Use `np.mean()` instead of a `for` loop to compute the average.
3. Show the average image, what we will call the "background" image.
4. Compute the absolute difference between the background image and the first frame of the video and show the result.
5. Use Otsu's method to determine a good threshold for the absolute difference image.
6. Threshold the absolute difference image to obtain a binary mask corresponding to the foreground pixels. Show the mask.
7. Now run the thresholding technique on each frame of the video and show the result as a video.
8. Each connected blob in the binary mask is a car. Put a bounding box around each car and show the result as a video.

Given a binary mask `fg` , here is an example of how to find the bounding boxes:

```
labels = skimage.measure.label(fg)
regions = skimage.measure.regionprops(labels)
for props in regions:
    minr, minc, maxr, maxc = props.bbox
```

Report

Your report should include the following:

Code explanation

Briefly explain your solution and any design choices you made that weren't specified in the instructions.

Clearly describe any external sources that were used (e.g. website or AI tools) and how they were used.

Discussion questions

1. When you compute the average image (background image), the cars have disappeared -- why?
2. How well does the technique work to separate the cars from the background? Where does it fail and why?

Submission instructions

Submit your Python script (.ipynb file) and report (PDF or docx). Please do not put them in a zip file; just submit the files directly.